ARMY RESEARCH LABORATORY

# Automatic Target Recognition Using a Modular Neural Network

by Lin-Cheng Wang, Sandor Der, and Nasser M. Nasrabadi

ARL-TR-1659 May 1998

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

**Army Research Laboratory**
Adelphi, MD 20783-1197

**ARL-TR-1659** **May 1998**

# Automatic Target Recognition Using a Modular Neural Network

Lin-Cheng Wang
State University of New York at Buffalo

Sandor Der and Nasser M. Nasrabadi
Sensors and Electron Devices Directorate, ARL

## Abstract

A modular neural network classifier has been applied to the problem of automatic target recognition (ATR) of targets in forward-looking infrared (FLIR) imagery. The classifier consists of several independently trained neural networks operating on features extracted from a local portion of a target image. The classification decisions of the individual networks are combined to determine the final classification. Experiments show that decomposition of the input features results in performance superior to a fully connected network in terms of both network complexity and probability of classification. The classifier's performance is further improved by the use of multiresolution features and by the introduction of a higher level neural network on top of the expert networks, a method known as *stacked generalization*. In addition to feature decomposition, we implemented a data decomposition classifier network and demonstrated improved performance. Experimental results are reported on a large set of FLIR images.

# Contents

# Figures

# Tables

# 1. Introduction

An automatic target recognizer (ATR) is an algorithm that locates potential targets in an image and identifies their types. Most ATR designs consist of several stages [1]. At the first stage, a target detector, operating on the entire image, detects some potential targets. In order to reduce the false-alarm rate, the second stage attempts to reject false targetlike objects (clutter) and retain targets. At the third stage, a set of features is computed; the selected features must effectively represent the target image. The fourth stage classifies each target image into one of a number of classes by using the features calculated at the previous stage. This report focuses on the last two stages: feature extraction and classification.

Target recognition using forward-looking infrared (FLIR) imagery of natural scenes is difficult because of the variability of target thermal signatures. Collected under different meteorological conditions, times of day, locations, ranges, etc, target signatures exhibit dramatic differences in appearance. Moreover, partial target obscuration and the presence of targetlike objects in the background make target recognition unreliable. Because of the high variability of target signatures, large FLIR data sets are required if statistical learning algorithms are to generalize well [2].

The algorithm described in this report seeks only to recognize targets that have been indicated by a detector operating on an entire image. The detector is assumed to have a position error of $\pm 5$ pixels rms. The range is assumed to be known, with an error of $\pm 10$ percent. This is a reasonable assumption in most instances, because infrared images will typically be acquired with information on the range to the center of the field of view and the depression angle of the sensor; this information allows a flat-earth approximation of the range to each pixel to be calculated. No assumption is made about the pose of the target; the image set contains targets from all aspect angles and from elevation angles between 0 and 10°. The classifier is trained and tested on all poses together.

Neural networks (NNs) have been recognized as powerful tools for solving pattern recognition problems. There is a close relationship between NN models and statistical pattern recognition methods. Various applications of NN technologies for ATR have been surveyed by Roth [3]. NNs can also be applied to extract essential features that provide a complete image description with lower dimensionality, as described by Daugman [4]. Casasent

and Neiberg [5] designed a feature space trajectory classifier NN to detect the position of each target and to classify its target class, where the feature set is a fixed set of wedge-sampled magnitude-squared Fourier transform features. Hecht-Nielson and Zhou [6] describe an NN target classifier that used a multiresolution Gabor local spatial frequency feature set.

A key advantage of using learning algorithms such as NNs is that since the algorithms automatically learn the available data, the resulting high-performance algorithms are tailored to the available data. In many applications, NNs give performance similar to that of learning vector quantizers (LVQs), $k$ nearest-neighbor algorithms, and other learning algorithms, but with significantly reduced computational complexity.

The difficulties associated with NNs include dimensionality and generalization problems. An NN requires, as a rule of thumb, approximately five data samples per free parameter. For learning many-dimensional data such as images, either a huge number of samples is required, or the data must first be reduced by the extraction of a relatively small number of relevant features for the NN to learn. If the number of data samples available is too small in comparison to the free parameters of the network, then the network will overfit the training data, resulting in poor generalization on test data. If the network has too few free parameters in comparison to the complexity of the classification problem, performance will be much lower than the theoretically achievable limit. Hence, designing an NN requires experimentation to determine the best architecture for a given problem and associated data set.

The rather complex task of ATR classification can be tackled through a modular approach, where each module is optimized to perform a relatively simple task in a complex overall operation. (Modularity has been found to be characteristic of the human visual system [7,8]).The modular NN classifier technique investigated in this report attempts to overcome the problems mentioned above.

First, the features are spatially divided into groups for classification by individual networks (committee members), and the outputs of the networks are then combined by a separate NN that forms the final decision. (Fig. 6 in sect. 3.3.2 shows the atchitecture of this type of classifier.) This decomposition of features has the advantage of greatly reducing the free parameters of each network, while keeping the number of data samples the same. The disadvantage is that features applied to separate expert networks do not interact with each other. Our experiments show that for our problem, the advantages of decomposing the features in this manner outweigh the disadvantages.

Second, we separate the set of all poses of all targets into groups based on the size and shape of target poses, and use individual NNs (experts) to learn each group, and a separate NN, called a gating network, to choose to which group an input data sample belongs. (Fig. 10 in sect. 4.3 shows the architecture of this classifier.) One advantage of this step is that it simplifies the classification problem that each expert network must solve. Another advantage is that knowledge of the size and shape of the hypothesized object allows tight windows to be drawn around the proposed object, which eliminate features from the background. One disadvantage is that the amount of training data for each expert network is reduced (on average) by a factor of the inverse of the number of expert networks. Another disadvantage is that the error in the gating network compounds any error in the expert networks. Our experiments, however, show that overall performance is improved by this decomposition-of-data strategy. We combine the two strategies—decomposition of features and decomposition of data—by forming each expert network using a set of modular networks that use feature decomposition. This architecture is somewhat similar to that used by Jacobs and Jordan [9].

This modular classifier has been trained and tested on the U.S. Army sponsored Comanche imagery set. This data set contains 10 military ground vehicles viewed from a ground-based second-generation FLIR sensor. The targets are viewed from arbitrary aspect angles, which are recorded in the ground truth (rounded to the nearest 5°). The images contain cluttered backgrounds and some partially obscured targets. The target signatures vary greatly, because portions of the imagery were collected at different times of day and night, at different locations (in Michigan, Arizona, and California), at different seasons, under varying weather conditions, and in different target exercise states. Examples of target chips for the 10 military ground vehicles at a 90° viewing angle are shown in figure 1.

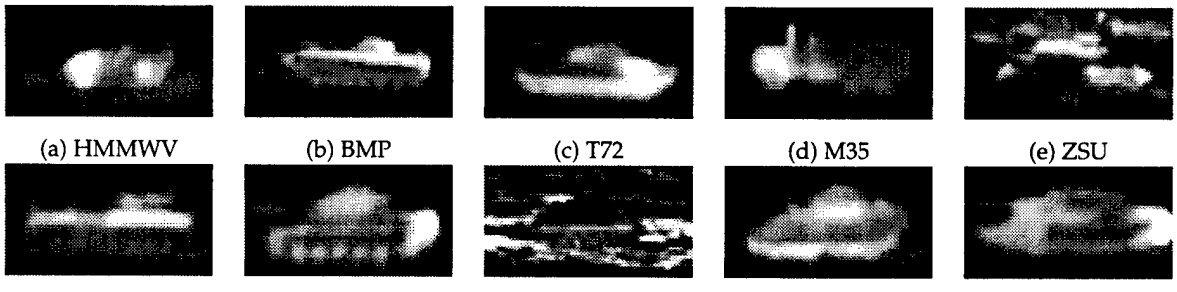



Figure 1. Examples of 10 target chips at 90° viewing angle.

In the remainder of this report, we describe the committee-of-networks and mixture-of-experts concepts (sect. 2) and the implementation of the committee of networks (sect. 3). We then present the implementation of the mixture-of-experts architecture and experimental results (sect. 4) and describe plans for future work (sect. 5).

# 2. Modular Neural Networks

In this report, we apply two types of modular NNs: a mixture-of-experts modular network (MEMN) [10], which performs decomposition of data, and a committee-of-networks modular network (CNMN) [11], which performs decomposition of features.

## 2.1 Mixture-of-Experts Modular Network (MEMN)

An MEMN consists of several *expert* NNs (modules), where each expert is optimized to perform a particular task of an overall complex operation. An integrating unit, called a *gating network*, is used to select or combine the outputs of the expert NN modules to form the final output of the MEMN.

In statistics, this type of modular approach is called the mixture model [12]. Mixture models have been used in a wide variety of applications where data derived from two or more categories are mixed in varying proportions [12]. A mixture model can be represented by a linear combination of outputs of several distinct functions $v_{k,m}(\cdot)$, $k = 1, 2, \ldots, K$, where $k$ is a label representing an individual NN, and $m$ is the class label:

$$w(m) = \sum_{k=1}^{K} \pi_k \, v_{k,m}(x), \tag{1}$$

where

$$0 \leq \pi_k \leq 1, \qquad k = 1, 2, \ldots, K, \tag{2}$$

and

$$\sum_{k=1}^{K} \pi_k \; = \; 1. \tag{3}$$

The parameters $\pi_k$ are called mixing weights, and the functions $v_{k,m}(\cdot)$ are called mixing components, where $k = 1, 2, \ldots, K$. Each function $v_{k,m}(\cdot)$ is responsible for classifying data sample $x$ in its corresponding category $k$, while the parameter $\pi_k$ is responsible for choosing the category $k$ by applying weights to the functions $v_{k,m}(\cdot)$. If the output of one function can accurately classify a data sample, its corresponding weighting factor should be high. Conversely, if the output of one function is a poor classifier, its corresponding weighting factor should be low.

According to the universal approximation theorem [13], a single NN with a sufficient number of hidden nodes and hidden layers (a sufficient complexity) can be optimized to any desired degree of accuracy for any problem (that is, it can be used as a universal approximator). However, this theorem does not provide any information about how the network can be designed optimally in terms of learning time and ease of implementation. If the task can be naturally decomposed into a set of simpler functions and the decomposition can be discovered, we can solve this task easily by using a modular network. In our modular NN classifier, the classification function is partitioned into a set of simpler functions, each simpler function is independently solved by an individual NN module (expert), and then the system obtains the final outcome by combining the outputs of all the individual modules (experts). Moreover, an asymptotical performance analysis for a modular NN shows that its performance is always equal to or better than that of a single NN [14] for sufficiently large training sets. Our experimentation supports the analysis: not only does it reduce learning time, but also our modular NN classifier significantly improves performance.

## 2.2 Committee-of-Networks Modular Network (CNMN)

The output of a committee of networks with input vector $x$ can be an average of the outputs of several distinct functions $u_{k,m}(\cdot)$, where $k = 1, 2, \ldots, K$ is the label representing the individual NN and $m$ is the class label:

$$v(m) = \frac{1}{K} \sum_{k=1}^{K} u_{k,m}(x_k), \tag{4}$$

where $x_k$ is the subset of the feature vector $x$ that is applied to network $k$. A more general form of a committee of networks is expressed by

$$v = \phi(u_1, u_2, \ldots, u_K), \tag{5}$$

where $\phi(\cdot)$ is a nonlinear function with inputs of $u_k$, $k = 1, 2, \ldots, K$. Each function $u_{k,m}(\cdot)$ is responsible for classifying data $x_k$ into its class $m$.

In the committee of networks, each member network receives distinct inputs, which are features extracted from one receptive field. That is, the input of a member network in the committee of networks is a subvector (an input data vector is partitioned). The reasons for partitioning are to provide noise immunity to other member networks, and to reduce the dimensionality. The final output is contributed by all the members. The total number of data vectors for training a member network is the total number of target chips ($N$). The stacked generalization (SG) NN [15] integrates the outputs of member networks in a nonlinear way. However, the nonlinear weighting

of an SG NN is fixed for every target chip. The output of an SG NN is not directly related to the input data.

In the MEMN, each expert receives the same input features, but each expert is trained to be optimized for a particular subset of data vectors. The number of data vectors used for training an expert network is on average $\frac{1}{N}$. The gating network is trained to select or combine the outputs of expert networks in order to form the final output. The gating network provides linear weighting for outputs of expert networks. Note that the weighting is not fixed for all target chips: it is a direct function of each data sample.

If the expert networks in an MEMN are combined by an SG NN for a particular data vector, only one of the expert networks can provide accurate classification, since this expert network is trained to be optimized. Another expert network would not respond appropriately, since it would be out of its training subspace. Thus, only $\frac{1}{N}$ of inputs to the SG NN are meaningful for this target chip. We would not expect a reliable output from the SG NN.

# 3. Design of *Expert Classifier* using *Committee of Networks*

## 3.1 Multiresolution Feature Extraction

In real-world FLIR images, the target is often partially occluded; thus, features whose domain consisted of all the target pixels would be unreliable. In this report, a feature extraction technique called the interest operator [16], is used to find the directional variances in the horizontal, vertical, and both diagonal directions for each block in target images, as shown in figure 2. These directional variances show the local activity in a block. The areas with high variance, such as edges or corners on a target, characterize a specific type of target. In FLIR imagery, there are some low gradient edges or corners on a target, which are also important for target classification. Thus, we can enhance the feature set by applying the interest operator again on subsampled versions of target images. We do not employ these directional

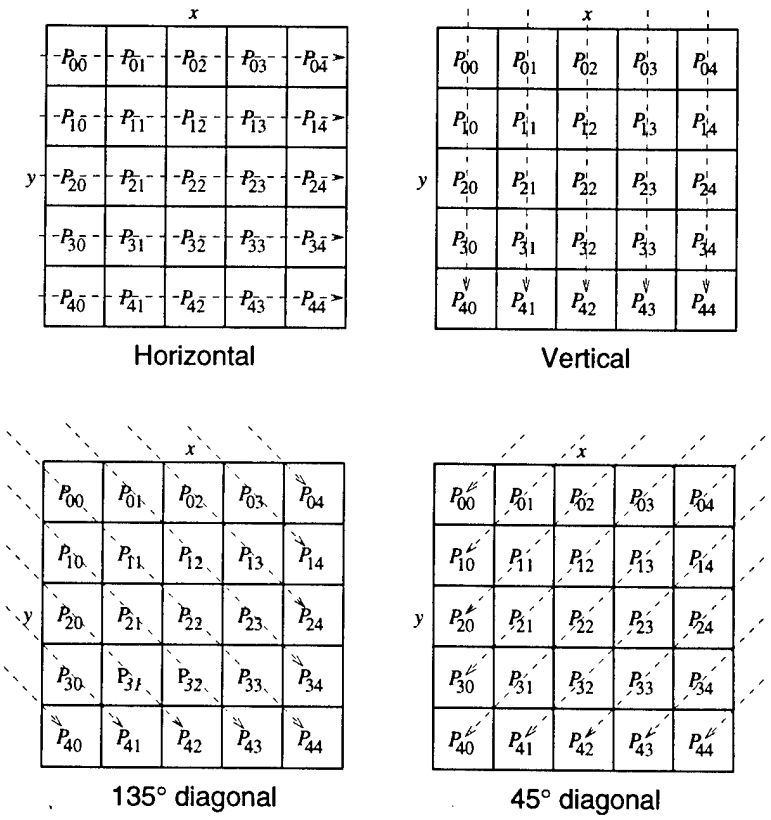



Figure 2. Feature extracted by interest operator.

variance features to form target templates for target recognition, as model-based ATR algorithms do [17]. Instead, we present these directional variance features to a neural-network-based classifier.

The following are equations for calculating the block variance, represented as $\sigma$, and the directional variances in horizontal, vertical, 135° diagonal, and 45° diagonal directions, respectively, represented as $\sigma_H$, $\sigma_V$, $\sigma_{D_{135°}}$, and $\sigma_{D_{45°}}$:

$$\mu = \frac{1}{N \times N} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} p(y,x), \tag{6}$$

$$\sigma = \frac{1}{N \times N} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} [p(y,x) - \mu]^2, \tag{7}$$

$$\sigma_H = \frac{1}{N \times N} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} [p(y,x) - p(y,x-1)]^2, \tag{8}$$

$$\sigma_V = \frac{1}{N \times N} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} [p(y,x) - p(y-1,x)]^2, \tag{9}$$

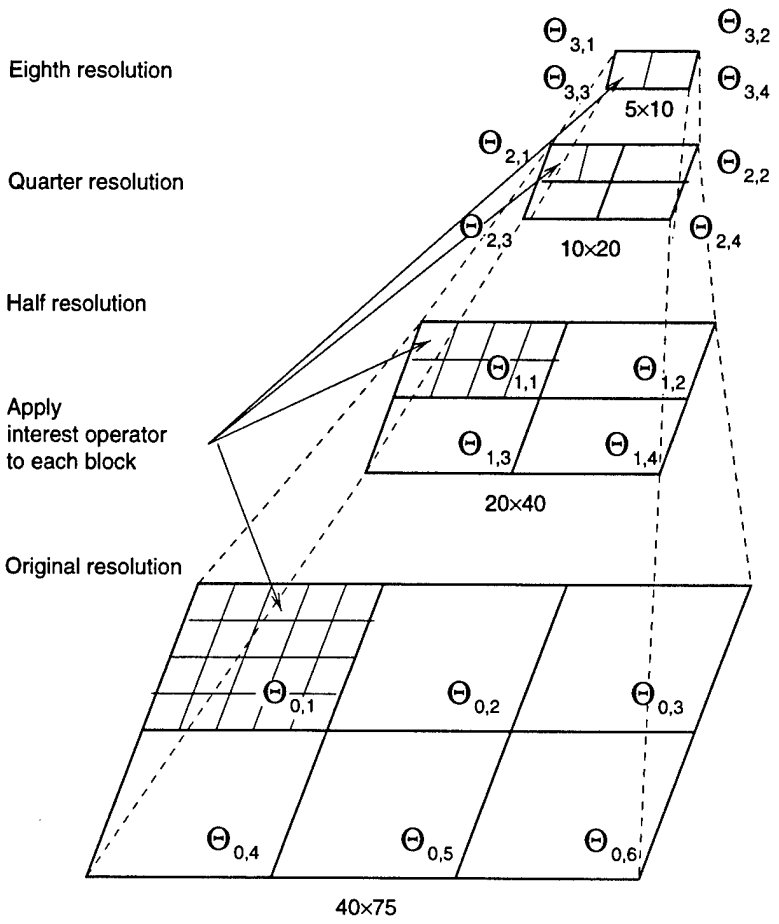
$$\sigma_{D_{135°}} = \frac{1}{N \times N} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} [p(y,x) - p(y-1,x-1)]^2, \tag{10}$$

$$\sigma_{D_{45°}} = \frac{1}{N \times N} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} [p(y,x-1) - p(y-1,x)]^2, \tag{11}$$

where $\{p(y,x), 0 \leq y,x \leq N-1\}$ represents the pixels in an $N \times N$ block ($N = 5$ in this report). We represent this variance feature set for an input image block as $\Theta$, where $\Theta = \{\sigma, \sigma_H, \sigma_V, \sigma_{D_{135°}}, \sigma_{D_{45°}}\}$.

The target image, sized 40 × 75, is partitioned into 120 nonoverlapping contiguous blocks of 5 × 5 pixels. We group these 120 image blocks into six receptive fields by location, as shown in figure 3. Thus, each receptive field has 20 image blocks. We apply the interest operator to each image block to obtain the directional variance features. The set of directional variances for each receptive field is represented as $\Theta_{0,i}$, where subscript 0 stands for original resolution and $i = 1, 2, \ldots, 6$ represents the six receptive fields. We refer to the image with original resolution as $L_0$. Multiresolution features are extracted from subsampled versions of targets. After the target image is downsampled three times, three new subsampled images are produced: a 20 × 40 image with half the resolution, $L_1$; a 10 × 20 image with a quarter the resolution, $L_2$; and a 5 × 10 image with an eighth the resolution, $L_3$. We partition these subsampled images into 32, 8, and 2 nonoverlapping

Figure 3.
Multiresolution feature extraction.




contiguous blocks, respectively, and group them into four receptive fields by location. Since the lowest resolution image has only two blocks, the two blocks are shared with the four receptive fields. The same interest operator is applied to these subsampled images. The directional variances of each receptive field on these two subsampled images are represented as $\Theta_{1,j}$, $\Theta_{2,j}$, and $\Theta_{3,j}$, where 1, 2, and 3 stand for lower resolutions and $j = 1, 2, \ldots, 4$ represents the four receptive fields at lower resolutions. The result of the interest operator on target images is shown in figure 4.

## 3.2 Neural Network Architecture and Training

The classification of each receptive field of a target image (*at different resolutions*) is performed by an NN classifier. The NNs implemented in this report are all multilayer perceptrons (MLPs) with one hidden layer, which use the standard backpropagation learning algorithm. The inputs of each MLP are the block variances and the directional variances. The MLPs have one output for each of the 10 target classes. The desired response of each

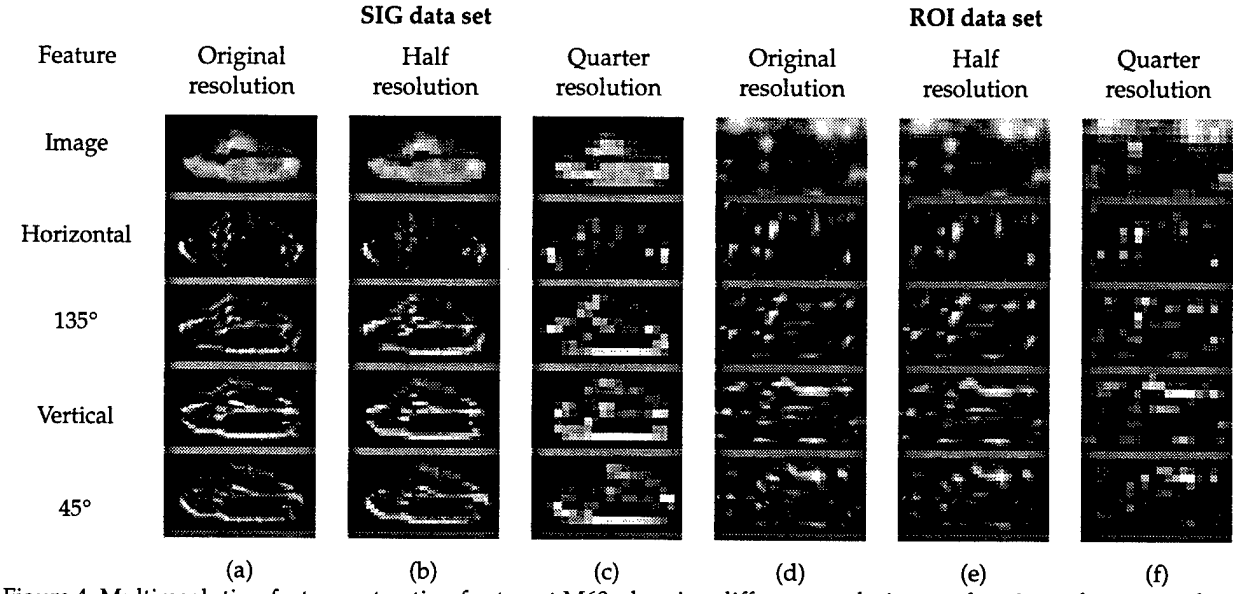



Figure 4. Multiresolution feature extraction for target M60, showing different resolutions and variance features in four directions.

MLP output during training was determined from the ground truth. The desired output of the correct target class was set to one, and the remaining desired outputs were set to zero. Because the backpropagated error derivative was set to the mean square error, the MLP outputs would converge to the a posteriori probabilities, given a sufficiently large training set [2].

The modular network classifier using multiresolution features described in this report is realized in two stages. At the first stage, we train individual expert networks, which receive input features from receptive fields at different resolutions. The NN weight that produces the best classification on the testing set by each expert network is selected for further integration. At the second (last) stage, the SG method is applied. All the expert networks are combined by a top-level NN (an SG MLP). The training of the SG MLP is based on the outputs of the expert networks.

## 3.3 Committee-of-Networks Modular Network (CNMN)

Computer simulations were performed to evaluate the performance of the CNMN classifier. This classifier has been trained and tested on the U.S. Army Comanche data set. We used a set of 10,000 target images for training the NNs, called the SIG training set, and another set of 1000 target images (disjoint from the training data) for testing, called the SIG testing set. Another data set (a set of 3460 target images), called ROI, which is considered difficult to classify because the images include complicated backgrounds and target obscuration, was used as a generalization data set. The

SIG training data set was used exclusively for training. After every epoch of training, we measured the performance of the algorithm on the SIG testing set to determine when to stop the training. We used the more difficult generalization set ROI to determine if the best stopping point was constant for sets of varying difficulty. Figure 4 shows target M60 taken from the SIG as well as the ROI data sets. It can be seen that the ROI target has a complicated background and is cluttered. Table 1 shows the complexity of each classifier described in this report and presents the performance of each by probability of correct classification for the SIG training set, the SIG testing set, and the ROI testing set. All the classifiers converged to quite consistent performance numbers regardless of initial weight values.

### 3.3.1 Fully Connected Network Classifier (FCNC)

In order to demonstrate the effectiveness of the committee-of-networks classifiers, we implemented a classifier (a single MLP) that receives feature inputs from whole images:

$$Y_{FCNC} = \Phi(\Theta_{0,1}, \Theta_{0,2}, \ldots, \Theta_{0,6}), \tag{12}$$

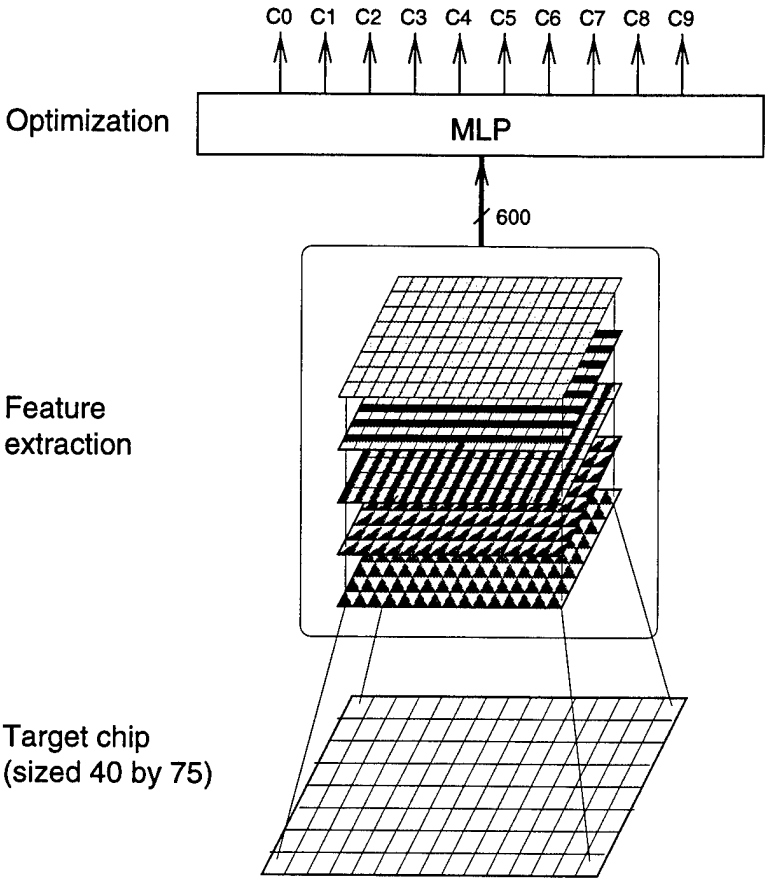
where $\Phi(\cdot)$ performs the NN function as a classifier. This MLP has 600 (= 120 $\times$ 5) inputs. The architecture of this fully connected network classifier (FCNC) is shown in figure 5. The number of hidden neurons in the MLP classifiers was optimized empirically and determined to be 200. (We accomplished this by training the algorithm with different numbers of hidden nodes, measuring the performance after training was completed, and choosing the best performing classifier.) Obviously, this MLP has too much

Table 1. Classification results.

| | Probability of correct classification (%) | | | |
|---|---|---|---|---|
| Method | Train SIG | Test SIG | Test ROI | Complexity |
| FCNC | 89.63 | 82.70 | 60.78 | 122,000 |
| OMNC | 91.46 | 85.70 | 65.84 | 33,000 |
| OMNC+SG | 92.52 | 87.90 | 67.40 | 35,100 |
| LMNC | 87.35 | 82.90 | 58.21 | 7,800 |
| MMNC | 92.91 | 88.30 | 67.28 | 40,800 |
| MMNC+SG | 94.50 | 91.50 | 69.39 | 49,600 |

FCNC = fully connected network classifier (sect. 3.3.1); OMNC = original-resolution modular network classifier (sect. 3.3.2); SG = stacked generalization; LMNC = lower resolution modular network classifier (sect. 3.3.3); MMNC = multiresolution modular network classifier (sect. 3.3.4).

Figure 5. A fully connected network classifier for ATR.




network capacity for our limited training set. The MLP exhibits a probability of correct classification of 89.63 and 82.70 percent for the SIG training and testing sets, respectively. There is a 7-percent difference in the probability of correct classification, even though both data sets are considered to be similar. This implies that the FCNC has a poor generalization capability. We also observe that the probability of correct classification for the ROI testing set is only 60.78 percent.

### 3.3.2 Original-Resolution Modular Network Classifier (OMNC)

In the second experiment, we implemented a committee-of-networks classifier with six individual MLPs, each of which receives feature inputs from only its corresponding receptive field. The inputs of each MLP are the block variances and the directional variances in four directions of 20 image blocks in a receptive field; hence, each MLP has 100 inputs. The number of hidden neurons in the MLP classifiers was optimized empirically, and determined to be 50. The MLPs have one output for each of the 10 target classes. Each MLP was trained independently and in parallel. The outputs of the MLPs

were combined by summing [11]:

$$P_i = \Phi(\Theta_{0,i}) \qquad i = 1, 2, \ldots, 6, \tag{13}$$

$$Y_{OMNC} = \sum_i P_i, \tag{14}$$

where $\Phi(\cdot)$ performs the NN function as a classifier.

This committee-of-networks classifier, which we call the original-resolution modular network classifier (OMNC), results in a probability of correct classification of 91.46, 85.70, and 65.84 percent for the SIG training set, the SIG testing set, and the ROI testing set, respectively. The OMNC outperforms the FCNC by 3.0 percent for the SIG testing set and 5.1 percent for the ROI testing set, but requires only 27 percent of the network capacity. In our experiment, the individual MLPs in the committee make correct classifications and show a high confidence level when their corresponding receptive fields are clear, while the individual MLPs make wrong classifications and show a low confidence level when their corresponding receptive fields are obscured. (We define the *confidence level* of the classification as the output margin between the largest and the second largest outputs.) By summing the outputs of the MLPs, the committee can tolerate some corrupted portions of a target image.

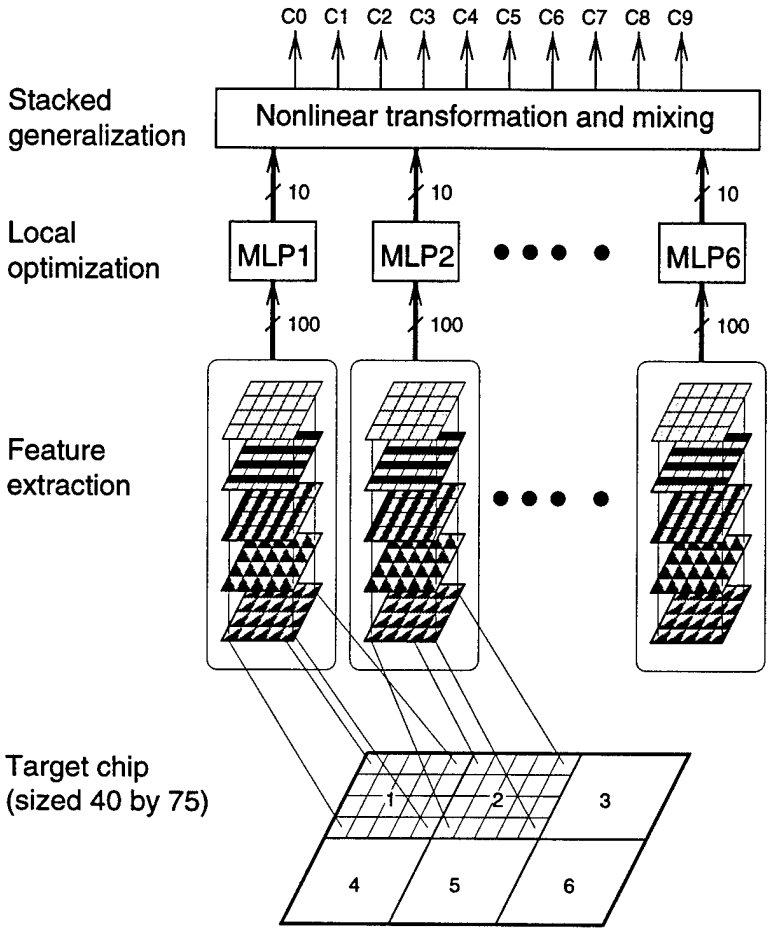
We also implemented a similar committee-of-networks classifier that used an MLP for each directional variance feature taken over the whole image (five MLPs of 120 input features each). The performance was only 56.23 percent on the ROI testing set, even though the performance of individual committee members was similar to that of the local receptive field committee. The superior performance of the committee of networks that split up the features by local receptive field is not surprising, as the a posteriori probabilities estimated by the individual committee members should display more statistical independence because of spatial separation.

Without excessively increasing network capacity, we can apply the SG method to combine the outputs of the MLPs. A higher level MLP is trained to learn the best combination of lower level MLPs:

$$Y_{OMNC+SG} = \Psi(P_1, P_2, \ldots, P_6), \tag{15}$$

where $\Psi(\cdot)$ represents the NN function for SG. The architecture of this committee of networks is shown in figure 6. The probabilities of correct classification for the SIG training set, the SIG testing set, and the ROI testing set are 92.52, 87.90, and 67.40 percent, respectively. As expected, SG yielded somewhat better performance than summing. Actually, SG improves not only the performance of the committee-of-networks classifier, but also the

Figure 6. A committee-of-networks classifier with feature extraction in original resolution.




generalization capability of networks. The difference in the probability of correct classification between the SIG training set and the SIG testing set is reduced to 4.62 percent.

### 3.3.3 Lower Resolution Modular Network Classifier (LMNC)

The first and second experiments use only single-resolution feature extraction (highest resolution). The single-resolution committee-of-networks classifier shows a satisfactory performance in terms of the probability of correct classification and the network capacity. In our third experiment, we designed a committee-of-networks classifier using features extracted from target images at lower resolutions. By reducing the resolution of targets, we can exploit the characteristics of directional variance features at different resolutions. In addition, we have fewer features extracted, so we can further reduce the capacity of the networks.

This classifier, which we call the lower resolution modular network classifier (LMNC), consists of four individual MLPs, where each MLP receives features extracted from only two subsampled targets on its corresponding receptive field. The outputs of the MLPs were combined by summing (note that the network capacity of this committee-of-networks classifier is very small):

$$Q_j = \Phi_j(\Theta_{1,j}, \Theta_{2,j}, \Theta_{3,j}) \qquad j = 1, 2, \ldots, 4, \tag{16}$$

$$Y_{LMNC} = \sum_j Q_j. \tag{17}$$

This classifier results in probabilities of classification of 87.35, 82.90, and 58.21 percent for the SIG training set, the SIG testing set, and the ROI testing set, respectively. The performance of the LMNC is close to that of the FCNC, although the LMNC requires less than 7 percent of the network capacity required by the FCNC.

### 3.3.4 Multiresolution Modular Network Classifier (MMNC)

In our final experiment, we combine the NN members in both previous committee-of-networks classifiers by using SG. Thus, a multiresolution modular network classifier (MMNC) is obtained:

$$P_i = \Phi_i(\Theta_{0,i}) \qquad i = 1, 2, \ldots, 6, \tag{18}$$

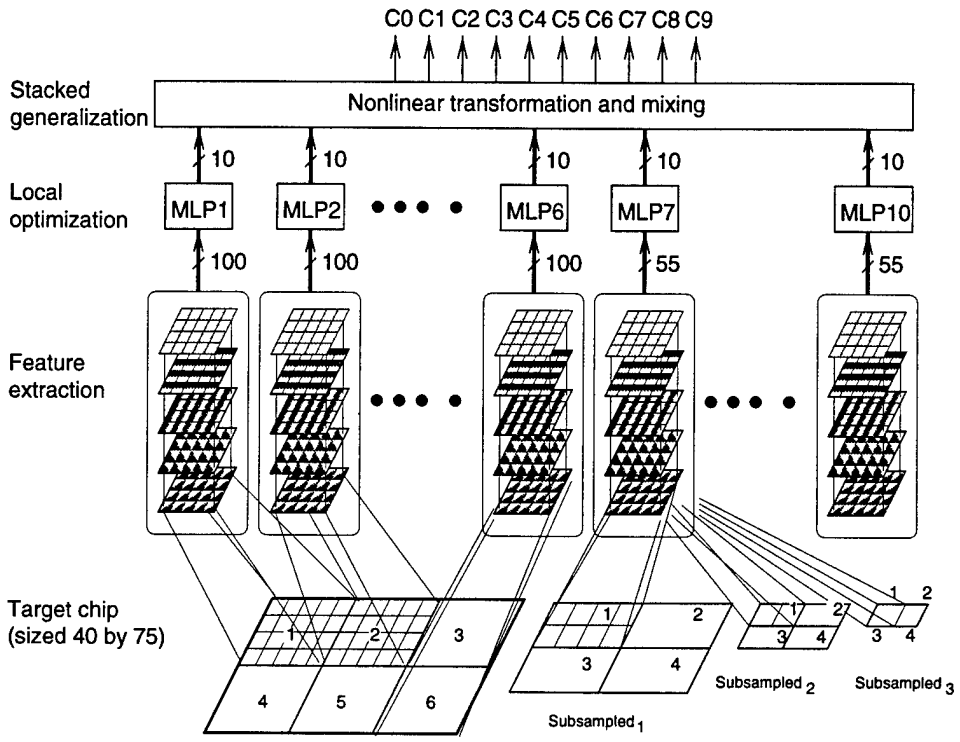
$$Q_j = \Phi_j(\Theta_{1,j}, \Theta_{2,j}, \Theta_{3,j}) \qquad j = 1, 2, \ldots, 4, \tag{19}$$

$$Y_{MMNC} = \Psi(P_1, P_2, \ldots, P_6, Q_1, Q_2, \ldots, Q_4). \tag{20}$$

The complete system is shown in figure 7. The stripes in the feature-extraction stage show the different block directional variances taken from a local region of the target chip and applied to the MLP. The 10 outputs of each of the local optimization MLPs are applied as inputs to the SG MLP.

By introducing more NN members associated with multiresolution features, this committee-of-networks classifier provides higher probabilities of correct classification for the SIG training set, the SIG testing set, and the ROI testing set: 94.50, 91.50, and 69.39 percent, respectively. Actually, SG not only improves the performance of the MMNC, but also overcomes the problem of the generalization capability. The difference in the probability of correct classification between the SIG training set and the SIG testing set is reduced to 3.00 percent.

Figure 7. A multiresolution committee-of-networks ATR classifier with stacked generalization.




### 3.3.5 Analysis of Classification done by Expert Networks and Modular Network

Table 2 shows the probability of correct classification given by each expert network in the committee-of-networks classifier MMNC. It can be seen from table 2 that not all expert networks classify target images well. The outcome is not surprising, because each expert network has input information from only a portion of a target image, which would not provide sufficient information for correct classification of most target images. Moreover, partial target obscuration degrades the classification performance. However, when we combine all the expert networks by summing the outputs, and/or by applying a higher level NN for SG, the effect is a dramatic increase in the probability of correct classification.

A high confidence level can be seen as classification certainty, while a low confidence level implies a doubtful outcome. In the implementation of a committee of networks, it is desirable that the expert networks produce a high confidence level when a correct classification is made, and a low confidence level when an incorrect classification is made. Thus, the incorrect classification by some expert network can be compensated for by another expert network with a high-confidence correct classification. Figure 8 shows the distributions of confidence levels for both correct and incorrect classifications for one of the expert networks ($P_2$) in the OMNC, one of the

Table 2. Classification performance of expert networks.

| | Correct classification (%) | | |
|---|---|---|---|
| Expert network | Train SIG | Test SIG | Test ROI |
| $P_1$ | 58.30 | 53.80 | 35.98 |
| $P_2$ | 78.56 | 73.10 | 51.68 |
| $P_3$ | 59.64 | 54.70 | 36.94 |
| $P_4$ | 64.85 | 60.00 | 32.77 |
| $P_5$ | 67.26 | 61.30 | 34.94 |
| $P_6$ | 67.22 | 59.60 | 37.34 |
| $Q_1$ | 70.94 | 67.50 | 43.99 |
| $Q_2$ | 69.25 | 64.60 | 46.82 |
| $Q_3$ | 69.67 | 64.70 | 36.59 |
| $Q_4$ | 68.36 | 61.80 | 39.05 |

expert networks ($Q_1$) in the LMNC, the MMNC, and the MMNC+SG. In figure 8(a), most of the correctly classified target images have high confidence levels, whereas most of the incorrectly classified target images have low confidence levels. Figure 8(b) has a similar distribution to that of figure 8(a), but the distribution for correct classification is not as sharp as in figure 8(a). This difference can be used to explain why the average probability of correct classification of expert networks in the LMNC is higher than that of expert networks in the OMNC, but the performance of the LMNC is nevertheless not as good as that of the OMNC when the outputs of expert networks are summed. There are fewer correctly classified target images with high confidence in the LMNC, which makes it hard to make up for the incorrectly classified target images. Obviously, in figure 8(c), the MMNC successfully reduces the number of incorrectly classified target images by summing the outputs of expert networks. Comparing figures 8(c) and (d), we observe that the SG method not only improves classification performance but also enhances classification confidence.

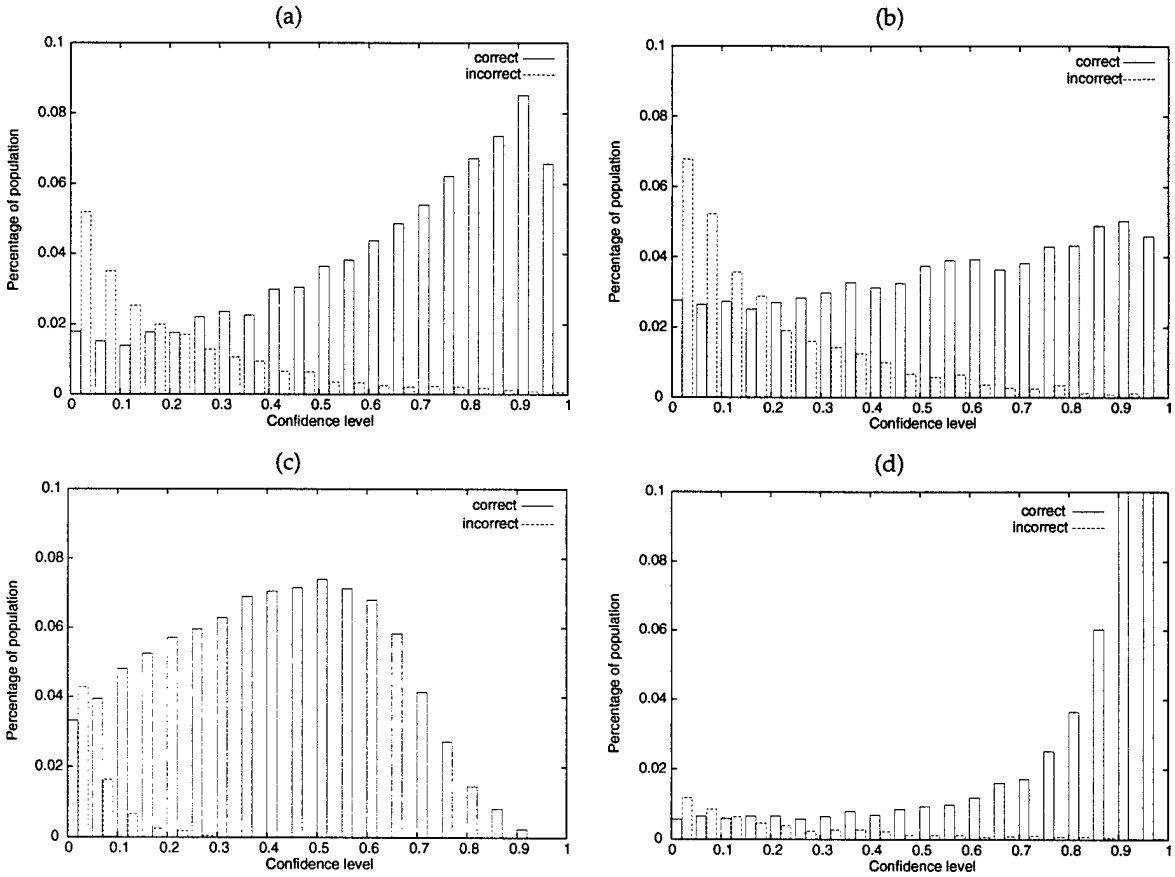



Figure 8. Distributions of confidence level for (a) expert network $P_2$, (b) expert network $Q_1$, (c) summing of 10 expert networks, and (d) stacked generalization.

# 4. Design of a *Target Classifier* using a *Mixture of Experts*

## 4.1 Formation of Windows (Categories)

One of the most important considerations in designing a modular network is keeping the number of experts to a minimum. Finding the optimum number of experts is related to the problem of statistical clustering of the training data. However, it is not feasible to perform clustering algorithms and to obtain a manageable number of clusters for a many-dimensional training data set, such as FLIR target chips. Fortunately, FLIR target chips (which are 3-D objects) can be projected to 2-D silhouettes. We can cluster the silhouettes of targets, rather than the target images. This is a suboptimal but computationally feasible solution. In our data set, there are 10 different vehicle targets, each having 72 projections (each aspect rounded to the nearest 5°), so the clustering is based on 720 silhouettes. We cluster the target chips into four categories that are based on the shapes of silhouettes. The union of the silhouettes of each cluster creates a *window*, which can cover all the targets belonging to the cluster. An advantage of this approach is that a large portion of the background pixels in a target image is removed from consideration.

The $K$-means algorithm [18] is used to perform the clustering of silhouettes. The input data set of the $K$-means algorithm consists of 720 silhouettes in the form of a binary image. The outputs are the bitmaps of the windows corresponding to each cluster and a mapping between individual silhouettes and the clusters. Figure 9 shows the clustering results for four windows. Table 3 shows the distribution of targets to each window. The four windows can be described as medium (0), small (1), large (2), and tall (3).

## 4.2 Background Removal Procedure

Natural FLIR target chips unavoidably contain pixels that fall on the nearby background of a target rather than on the targets themselves, because the chips are rectangular and correspond to the size of the largest targets. The information carried by background pixels is irrelevant for classifying targets and serves only to mistrain the classifier. Better classification performance can be achieved through background clipping. The $K$-means process described above serves to reduce the number of background pixels in each

Figure 9. Four windows and clipping and expanding operations.

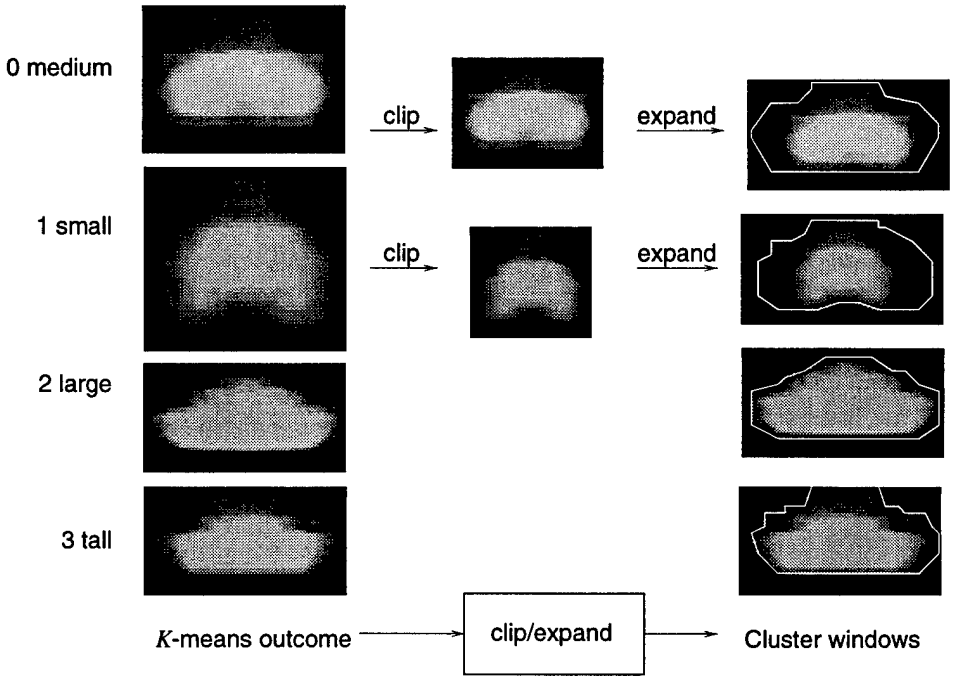



Table 3. Clustering results of four windows.

| | Cluster results | | | |
|---|---|---|---|---|
| Target type | Window 0 | Window 1 | Window 2 | Window 3 |
| HMMWV | 46 | 26 | 0 | 0 |
| BMP | 16 | 10 | 0 | 46 |
| T72 | 12 | 6 | 0 | 54 |
| M35 | 12 | 14 | 0 | 46 |
| ZSU | 8 | 14 | 0 | 50 |
| 2S1 | 8 | 10 | 37 | 17 |
| M60 | 6 | 6 | 50 | 10 |
| M113 | 58 | 14 | 0 | 0 |
| M3 | 8 | 10 | 34 | 20 |
| M1 | 4 | 6 | 50 | 12 |

window, because pixels that fall outside the union of the silhouettes in a given cluster can be eliminated as inputs to the neural net. Thus, the classification performed by the modular NN can be referred to as a segmentation-based classification method. Although silhouettes could more precisely separate the background from target chips that can windows, the huge number of silhouettes makes the selection of the exact silhouette difficult. The window of each cluster provides a suboptimal solution for background removal.

The background removal for each window is executed in three steps. The target chip is clipped to the smallest rectangle that includes all the pixels in the window. The small rectangle is zoomed to the size of the original target chip (small windows thus involve greater expansion than larger windows). The feature-extraction procedure described earlier is then applied to the new target chip.

## 4.3 Architecture and Neural Network Training

After assigning a window label to each of the training vectors, we can now design each individual expert classifier using only the subset of the training data that belongs to that particular window. All these expert classifiers are designed to classify 10 classes of targets, although some windows may not consist of all 10 target classes. We implemented each expert classifier by using a committee-of-networks classifer, described earlier, as shown in figure 4.

The gating network is also a committee-of-networks classifier. The gating network is designed so that each of its outputs estimates the probability that the input target chip belongs to the corresponding window. The number of output nodes of the gating network is 4. The system diagram of the complete system is shown in figure 10.

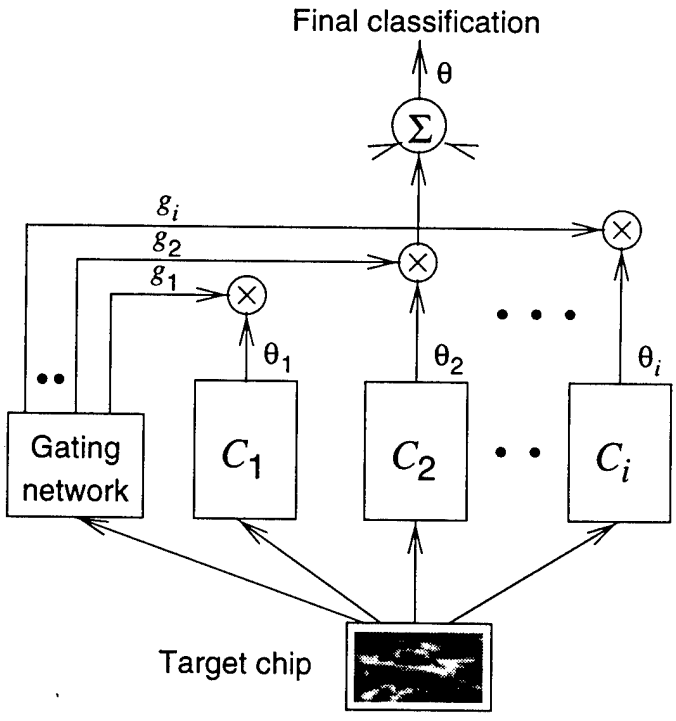



Figure 10. Block diagram of modular neural network classifier.

# 5. Experimental Results

## 5.1 Classification of Gating Network

The gating network is a committee-of-networks classifier with several member networks and an SG NN on the top of the member networks. The gating network is designed so that each of its outputs estimates the a posteriori probability of its corresponding window. The input data set is the entire training set, since data set partitioning is not needed for training the gating network. The inputs to each member network are the multiresolution features extracted from one local receptive field of the target chip. The number of output nodes of a member network is equal to the number of windows, which was set to 4 in the work reported here. The number of hidden nodes for each member network in the committee is 50, based on our experience designing the committee-of-networks classifier, discussed in section 3.3. The SG NN of the gating network receives the outputs of member networks as its inputs, and then calculates the final output of the gating network.

The results of the gating network performance alone are shown in tables 4 and 5. From the confusion matrix, we observe that the small window (Window 1) is easy to confuse with the medium window (Window 0), while Window 0 is in turn confused with the tall window (Window 3). Window 3 is confused with the large window (Window 2). In practice, the gating network is a shape classifier, which classifies target chips into one of the windows. The greatest confusion occurs between the tall and large windows, 3 and 2. Table 4 shows that the target chips belonging to these two windows are largely disjoint. Although combining these two windows into one might improve the window classification performance of the gating network, the largely disjoint membership of these two windows would reduce the target classification performance of the expert classifiers.

Window classification is an easier task than target classification, for a number of reasons. The windows were designed by the $K$-means algorithm to maximize the separation between windows, and thus minimize the difficulty in classifying windows. Also, there are only four windows, rather than 10 targets. The window classifier does suffer from a disadvantage. Because the window grouping was based entirely on target silhouettes, and because individual windows contain several different target types with different internal signatures, it is reasonable to suppose that the window clas-

Table 4. Confusion matrix of window classification by gating network for SIG training data. Probability of correct classification is 89.42%.

| Window | 0 | 1 | 2 | 3 | Total |
|---|---|---|---|---|---|
| 0 | 1784 | 196 | 11 | 243 | 2234 |
| 1 | 105 | 1618 | 1 | 32 | 1756 |
| 2 | 11 | 6 | 2327 | 287 | 2631 |
| 3 | 115 | 27 | 66 | 3568 | 3776 |

Table 5. Confusion matrix of window classification by gating network for ROI test data. Probability of correct classification is 85.23%.

| Window | 0 | 1 | 2 | 3 | Total |
|---|---|---|---|---|---|
| 0 | 515 | 62 | 5 | 52 | 634 |
| 1 | 14 | 729 | 10 | 27 | 780 |
| 2 | 11 | 6 | 314 | 140 | 471 |
| 3 | 74 | 24 | 86 | 1391 | 1575 |

sifier must rely on the thermal gradient at the silhouette edge of the target to the exclusion of the internal thermal signature information. The silhouette information is more stable than the internal information, because of thermal signature variability due to environmental conditions (although the information content of the silhouette is obviously less than the information content of the silhouette and internal signature together). The relative performance of the window classifier on the SIG training and ROI testing sets, compared with the relative performance of the expert networks on the data sets, is instructive. The window classifier shows little decrease in performance between the training and testing sets, while the expert networks show much greater generalization problems, suggesting that a preponderance of the classification information is in the silhouette gradient.

## 5.2 Classification of Expert Classifiers

Results for the individual expert classifiers and the mixture-of-experts modular network (MEMN) classifier are summarized in table 6. To demonstrate the improvement performed by the MEMN classifier (following a divide-and-conquer strategy), we also show the classification results for the committee-of-networks modular network (CNMN) classifier in the four window categories. For the ROI testing set, each expert classifier outperforms the CNMN classifier for its category (window) by 12 to 19 percent. There are two reasons for the improvement. First, the CNMN classifier learns the entire target data set, which is a more difficult problem than learning a subset of it. Second, the windows allow the removal of extraneous background that can only confuse the classifier. To support these conclusions, we trained an expert classifier without clipping and expanding

Table 6. Classification results for all classifiers.

| Classifier | CNMN | | MEMN | |
|---|---|---|---|---|
| | Train SIG | Test ROI | Train SIG | Test ROI |
| 0 | 94.32 | 77.44 | 94.49 | 93.85 |
| 1 | 88.67 | 52.44 | 98.63 | 68.33 |
| 2 | 97.15 | 65.18 | 99.62 | 84.71 |
| 3 | 95.71 | 74.54 | 99.07 | 86.35 |
| Single | 94.75 | 68.81 | — | — |
| Ground truth | — | — | 98.15 | 83.44 |
| Oracle | — | — | 99.14 | 88.15 |
| Hard limited | — | — | 94.62 | 74.68 |
| Mixture | — | — | 95.49 | 75.58 |

target chips for category Window 1. This expert classifier gives performance results of 93.28 and 63.85 percent for the SIG training and ROI testing sets, respectively. It gives 5 and 10 percent improvement for both data sets when the mixture model is adopted, and 5 and 5 percent improvement for both data sets when clipping and expanding operations are adopted.

The expert classifier for classifying the category Window 1, the smallest window, has the worst probability of correct classification. The clipping and expanding operations do help by removing large areas of background. For example, the probability of correct classification for category Window 1 performed by its expert classifier improves around 10 and 16 percent for the SIG and ROI data sets. Hower, the effective target area for category Window 1 is small, so that the characteristics of individual targets are not easily distinguished.

## 5.3 Classification of Mixture-of-Experts Classifiers

As mentioned earlier, each individual expert classifier is trained on only the subset of the training data that belongs to its category, while the gating network is trained on all the data. The gating network is designed to obtain the final output of the mixture-of-experts modular network classifiers by linearly combining the outputs of the expert classifiers. The probabilities of correct window selection performed by the gating network are 89.89 and 85.23 percent for the SIG training set and the ROI testing set, respectively. The mixture-of-experts modular network classifier results in a probability of correct classification of 95.49, 90.53, and 75.58 percent for the SIG training set, the SIG testing set, and the ROI testing set, respectively. The mixture-of-experts modular network classifier shows improvement over the committee-of-networks classifier by around 7 percent for the ROI test-

ing set, but only 1 percent for the SIG training set. Tables 7 and 8 show the confusion matrices for the mixture-of-experts classifier on the training and testing sets, respectively.

For comparison, we have also combined the experts using a hard limited gating network. Instead of using a linear combination of the expert networks weighted by the gating network's output, we used only the output of that expert network which had the highest gating network output. Results are shown in table 6.

We also show the results for an "oracle" gating network, which selects the correct class if any of the individual expert classifiers do so. The results of the oracle serve as a theoretical upper bound for the MEMN classifier. Of course, there is no way to ensure that a gating network achieves this. Another result shown is based on a "ground-truth" gating network, which selects windows according to the ground truth. If the gating network were perfect and achieved 100-percent probability of correct window selection, the probabilities of correct target classification for the SIG training set and the ROI testing set would be 98.15 and 83.44 percent, respectively. On the other hand, the MEMN classifier has the potential to produce results that match the ground truth if the gating network is designed perfectly.

Table 7. Confusion matrix of target classification by modular neural network classifier for SIG training data. Probability of correct classification is 95.49%.

| Type | HMMWV | BMP | T72 | M35 | ZSU23 | 2S1 | M60 | M113 | M3 | M1 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| HMMWV | 964 | 0 | 0 | 4 | 14 | 0 | 1 | 2 | 3 | 2 | 990 |
| BMP | 11 | 986 | 5 | 2 | 4 | 11 | 1 | 1 | 10 | 4 | 1035 |
| T72 | 2 | 4 | 1103 | 2 | 4 | 3 | 10 | 2 | 3 | 8 | 1141 |
| M35 | 6 | 0 | 3 | 1021 | 3 | 0 | 2 | 3 | 2 | 2 | 1042 |
| ZSU23 | 3 | 0 | 2 | 1 | 1098 | 5 | 3 | 2 | 5 | 1 | 1120 |
| 2S1 | 15 | 40 | 9 | 1 | 2 | 1031 | 3 | 6 | 6 | 4 | 1117 |
| M60 | 3 | 4 | 7 | 4 | 18 | 2 | 1078 | 3 | 8 | 4 | 1131 |
| M113 | 12 | 3 | 4 | 6 | 4 | 2 | 1 | 618 | 4 | 11 | 665 |
| M3 | 3 | 16 | 7 | 2 | 3 | 10 | 3 | 8 | 1037 | 8 | 1097 |
| M1 | 5 | 12 | 9 | 4 | 7 | 1 | 8 | 9 | 12 | 992 | 1059 |

Table 8. Confusion matrix of target classification by modular neural network classifier for ROI testing data. Probability of correct classification is 75.58%.

| Target | HMMWV | BMP | T72 | M35 | ZSU23 | 2S1 | M60 | M113 | M3 | M1 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| HMMWV | 651 | 35 | 18 | 7 | 19 | 9 | 11 | 25 | 4 | 20 | 799 |
| BMP | 20 | 535 | 28 | 10 | 26 | 30 | 7 | 14 | 11 | 17 | 698 |
| T72 | 15 | 24 | 545 | 6 | 49 | 24 | 38 | 7 | 18 | 43 | 769 |
| M35 | 34 | 5 | 13 | 463 | 8 | 3 | 12 | 16 | 4 | 19 | 577 |
| M60 | 7 | 13 | 58 | 22 | 26 | 17 | 421 | 1 | 11 | 41 | 617 |

# 6. Conclusions and Future Work

This report demonstrates the improved performance that we obtain from using a combination of feature decomposition and data decomposition strategies for training modular NNs to classify ground vehicles in FLIR imagery. The data decomposition technique is particularly helpful for view-based vision applications, because of the reduction in extraneous background pixels that can be achieved by windowing.

Future work will concentrate on training the classifiers with novel bootstrapping techniques that form additional training samples from combinations of existing samples. Preliminary results suggest that these techniques will improve performance. In addition, performance could be improved by the application of a weight-cutting algorithm to the individual NNs.

## References


1. B. Bhanu, "Automatic target recognition: State of the art survey," *IEEE Trans. Aerospace Elect. Sys.*, 22(4), 364–379 (1986).

2. S. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Co. (1994).

3. M. W. Roth, "Survey of NN technology for automatic target recognition," *IEEE Trans. Neural Networks*, 1(1), 28–43 (1990).

4. J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust. Speech Signal Process.*, 36(7), 1169–1179 (1988).

5. D. P. Casasent and L. M. Neiberg, "Classifier and shift-invariant automatic target recognition neural networks," *Neural Networks*, 8(7–8), 1117–1129 (1995).

6. R. Hecht-Nielsen and Y.-T. Zhou, "VARTAC: A foveal active vision ATR system," *Neural Networks*, 8(7), 1309–1321 (1995).

7. J. C. Houk, "Learning in modular networks," in *Proc. Seventh Yale Workshop on Adaptive and Learning Systems*, 80–84 (1992).

8. D. C. Van Essen, C. H. Andersen, and D. J. Felleman, "Information processing in the primate visual system: An integrated systems perspective," *Science*, 255, 419–423 (1992).

9. R. A. Jacobs and M. I. Jordan, "Learning piecewise control strategies in a modular NN architecture," *IEEE Trans. Systems Man Cybernetics* 2(23), 337–345 (1993).

10. R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, 3, 79–87 (1991).

11. M. P. Perrone, "General averaging results for convex optimization," in *Proc. Connectionist Models Summer School*, 364–371 (1993).

12. D. M. Titterington, A.F.M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons (1985).

13. K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, 2, 359–366 (1989).

14. Lin-Cheng Wang, Sandor Z. Der, and Nasser M. Nasrabadi, "Asymptotical analysis of a modular network," *IEEE Int. Conf. Neural Networks 1997*, to appear (1997).

15. D. H. Wolpert, "Stacked generalization," *Neural Networks*, 5(2), 241–260 (1992).

16. H. Moravec, *Robot Rover Visual Navigation*, U.M.I. Research Press, Ann Arbor, MI (1981).

17. A. Kramer, D. Perschbacher, R. Johnston, and T. Kipp, "Relational template matching algorithm for FLIR ATR," *Proc. SPIE Aerospace Remote Sensing*, 29–37 (April 1993).

18. J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley (1991).

# Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Ofc of the Dir Rsrch and Engrg
Attn R Menz
Pentagon Rm 3E1089
Washington DC 20301-3080

Ofc of the Secy of Defns
Attn ODDRE (R&AT) G Singley
Attn ODDRE (R&AT) S Gontarek
The Pentagon
Washington DC 20301-3080

OSD
Attn OUSD(A&T)/ODDDR&E(R) R Tru
Washington DC 20301-7100

CECOM
Attn PM GPS COL S Young
FT Monmouth NJ 07703

CECOM
Sp & Terrestrial Commctn Div
Attn AMSEL-RD-ST-MC-M H Soicher
FT Monmouth NJ 07703-5203

Dept of the Army (OASA) RDA
Attn SARD-PT R Saunders
103 Army
Washington DC 20301-0103

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Prsnnl
Attn J Hiller
The Pentagon Rm 2C733
Washington DC 20301-0300

Dir of Assessment and Eval
Attn SARD-ZD H K Fallin Jr
103 Army Pentagon Rm 2E673
Washington DC 20301-0163

Hdqtrs Dept of the Army
Attn DAMO-FDT D Schmidt
400 Army Pentagon Rm 3C514
Washington DC 20301-0460

MICOM RDEC
Attn AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

US Army CECOM
Night Vision & Elec Sensors Dir
Attn AMSEL-RD-NV-VISPD C Hoover
Attn B Redman
Attn D N Barr
10221 Burbeck Rd, Ste 430
FT Belvoir VA 22060-5806

US Army Edgewood RDEC
Attn SCBRD-TD J Vervier
Aberdeen Proving Ground MD 21010-5423

US Army Info Sys Engrg Cmnd
Attn ASQB-OTD F Jenia
FT Huachuca AZ 85613-5300

US Army Materiel Sys Analysis Agency
Attn AMXSY-D J McCarthy
Aberdeen Proving Ground MD 21005-5071

US Army Matl Cmnd
Dpty CG for RDE Hdqtrs
Attn AMCRD BG Beauchamp
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Matl Cmnd
Prin Dpty for Acquisition Hdqrts
Attn AMCDCG-A D Adams
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Matl Cmnd
Prin Dpty for Techlgy Hdqrts
Attn AMCDCG-T M Fisette
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Mis Cmnd
Attn AMSMI-RD-MG-IP R Sims
Redstone Arsenal AL 354898-5253

US Army Natick RDEC
Acting Techl Dir
Attn SSCNC-T P Brandler
Natick MA 01760-5002

## Distribution (cont'd)

US Army NVESD
Attn AMSRL-RD-NV-UAB C Walters
Attn L Kruthers
10221 Burbeck Rd Ste 40
FT Belvoir VA 22060

US Army Rsrch Ofc
Attn G Iafrate
4300 S Miami Blvd
Research Triangle Park NC 27709

US Army Simulation, Train, & Instrmntn Cmnd
Attn J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army Tank-Automtv & Armaments Cmnd
Attn AMSTA-AR-TD C Spinelli
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Tank-Automtv Cmnd
Rsrch, Dev, & Engrg Ctr
Attn AMSTA-TA J Chapin
Warren MI 48397-5000

US Army Test & Eval Cmnd
Attn R G Pollard III
Aberdeen Proving Ground MD 21005-5055

US Army Train & Doctrine Cmnd
Battle Lab Integration & Techl Dirctrt
Attn ATCD-B J A Klevecz
FT Monroe VA 23651-5850

US Military Academy
Dept of Mathematical Sci
Attn MAJ D Engen
West Point NY 10996

Nav Surface Warfare Ctr
Attn Code B07 J Pennella
17320 Dahlgren Rd Bldg 1470 Rm 1101
Dahlgren VA 22448-5100

GPS Joint Prog Ofc Dir
Attn COL J Clay
2435 Vela Way Ste 1613
Los Angeles AFB CA 90245-5500

DARPA
Attn B Kaspar
Attn L Stotts
3701 N Fairfax Dr
Arlington VA 22203-1714

University of Texas
ARL Electromag Group
Attn Campus Mail Code F0250 A Tucker
Austin TX 78713-8029

Clark Atlanta Univ
Watervlet Analysis Group
Attn L Kaplan
Attn R Murenzi
James P Brawley Dr at Fair Stret SW
Atlanta GA 30314

College of Staten Island/CUNY
Dept of Engrg Sci & Physics
Attn S Rizvi
2800 Victory Blvd Rm 1N-223
Staten Island NY 10314

Georgia Inst of Techlgy
School of Electrical & Computer Engrg
Attn M Smith
Atlanta GA 30332-0150

Univ of Maryland
Dept of Elec Engrg
Attn Q Zheng
Attn R Chellappa
A V Williams Bldg Rm 2365
College Park MD 20742-3285

ERIM
Attn C Dwan
Attn D Zhao
Attn J Ackenhusen
Attn Q Holmes
1975 Green Rd
Ann Arbor MI 48105

Lockheed Martin Vought
Attn B Evans Stop L37-01
Attn D DuBois Stop LOST-20
Attn K Jenkins Stop L37-01
PO Box 650003
Dallas TX 7526-0003

## Distribution (cont'd)

Natl Inst Standards/Tech
Attn J Phillips
Bldg 225 Rm A216
Gaithersburg MD 20899

Palisades Inst for Rsrch Svc Inc
Attn E Carr
1745 Jefferson Davis Hwy Ste 500
Arlington VA 22202-3402

US Army Rsrch Lab
Attn AMSRL-CI-LL Techl Lib (3 copies)
Attn AMSRL-CS-AL-TA Mail & Records Mgmt
Attn AMSRL-CS-AL-TP Techl Pub (3 copies)
Attn AMSRL-SE J M Miller
Attn AMSRL-SE J Pellegrino

US Army Rsrch Lab (cont'd)
Attn AMSRL-SE-EE Z G Sztankay
Attn AMSRL-SE-SE D Nguyen
Attn AMSRL-SE-SE L A Chan
Attn AMSRL-SE-SE L Bennett
Attn AMSRL-SE-SE L-C Wang
Attn AMSRL-SE-SE M Lander
Attn AMSRL-SE-SE M Vrabel
Attn AMSRL-SE-SE N Nasrabadi
Attn AMSRL-SE-SE P Rauss
Attn AMSRL-SE-SE S Der (5 copies)
Attn AMSRL-SE-SE T Kipp
Attn AMSRL-SE-EI B Clark
Attn AMSRL-SE-E H Pollehn
Adelphi MD 20783-1197

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | May 1998 | Progress, June 1997 |

**4. TITLE AND SUBTITLE**
Automatic Target Recognition Using a Modular Neural Network

**5. FUNDING NUMBERS**
PE: 61102A

**6. AUTHOR(S)**
Lin-Cheng Wang (State University of New York at Buffalo), Sandor Der, and Nasser M. Nasrabadi (ARL)

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
U.S. Army Research Laboratory
Attn: AMSRL-SE-SE (sder@arl.mil)
2800 Powder Mill Road
Adelphi, MD 20783-1197

**8. PERFORMING ORGANIZATION REPORT NUMBER**
ARL-TR-1659

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**
AMS code: 611102.305
ARL PR: 7NE0M1

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

A modular neural network classifier has been applied to the problem of automatic target recognition (ATR) of targets in forward-looking infrared (FLIR) imagery. The classifier consists of several independently trained neural networks operating on features extracted from a local portion of a target image. The classification decisions of the individual networks are combined to determine the final classification. Experiments show that decomposition of the input features results in performance superior to a fully connected network in terms of both network complexity and probability of classification. The classifier's performance is further improved by the use of multiresolution features and by the introduction of a higher level neural network on top of the expert networks, a method known as *stacked generalization*. In addition to feature decomposition, we implemented a data decomposition classifier network and demonstrated improved performance. Experimental results are reported on a large set of FLIR images.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| ATR, FLIR, learning decomposition, object recognition | 43 |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102